phytron

Intelligent Powerpack IPP

Compact Stepper Motor Power Stage

with Integrated Axis Controller

and Power Supply

Preliminary Manual for IPP 92-70 and IPP 172-140

Preliminary Edition MA 1119-A001a

Contents

Page

1 Overview . . . 3

2 Technical Data . . . 4
2.1 Technical Data of the IPP's Stepper Motor Power Stage . . . 4
2.2 Mechanical Dimensions . . . 5
2.3 Interface Connectors, Diagnostic Displays . . . 6
2.3.1 Front Plate Layout . . . 6
2.3.2 Mains Connector . . . 7
2.3.3 Motor Connector . . . 7
2.3.4 Bus Interface . . . 7
2.3.4.1 RS 232 Interface . . . 8
2.3.4.2 RS 485-Interface . . . 9
2.3.5 Limit Switch Connector . . . 10
2.3.6 Digital Inputs and Outputs . . . 10
2.3.7 Voltage Levels . . . 11
2.3.8 Diagnostic LEDs . . . 13

3 Putting an IPP into Operation . . . 14
3.1 Things You Need for the First Test of an IPP . . . 14
3.2 First Steps with an IPP . . . 14

4 Test and Configuration of an IPP with the Software IPCOMM . . . 16
4.1 Overview . . . 16
4.2 Hardware Requirements to Use IPCOMM . . . 16
4.3 Installation of IPCOMM . . . 16
4.3.1 Installation of IPCOMM under Windows 3.x . . . 16
4.3.2 Installation of IPCOMM under Windows 95 . . . 17
4.3.3 Registration of Software Modules (Tools) . . . 17
4.4 First Test of IPCOMM with an IPP . . . 18

5 Overview over the IPCOMM Functions . . . 19

6 Reference List of the IPCOMM Functions . . . 21
6.1 Parameter Definitions and the Online Operation . . . 21
6.2 Status Display . . . 27
6.3 Ramp Programming . . . 28

7 IPP Operation controlled by an PLC . . . 30
7.1 Interface Definition in Basic PLC Mode . . . 30
7.2 Programming of PLC Functions . . . 32
7.3 Hints for Turning the PLC Mode into Operation . . . 33

8 Problem Solving . . . 34

9 ESD-Protection Measures . . . 35

10 Quality Management System . . . 35

11 Options . . . 36
11.1 Step Failure Indicator Module SFI . . . 36
11.2 Analog/Digital Converter Extension . . . 36
11.3 Digital I/O Extension . . . 36

1 Overview

Phytron's "Intelligent Powerpack" IPP is a compact device which contains in one single housing all you need to drive a 2-phase stepper motor.

The main building blocks of the IPP are:

- A high efficient power stage for driving bipolar 2-phase stepper motors
- An axis controller (indexer) to control the power stage (step resolution: ministep with 1/8th full step, controlled by a microcontroller)
- A serial interface and digital inputs and outputs, compatible with most PLCs
- A power supply for direct connection to mains voltage 230 V_{AC} and 115 V_{AC} (option).

Two different power ranges are available:
The IPP 92-70 delivers up to 9 A current with 70 V nominal motor voltage,
the IPP 172-140 up to 17 A current with 140 V.

The combination of axis controller and power stage makes the IPP devices very flexible and well suited for many of applications:

- Dynamical adaption of step resolution to the speed of rotation:

  If the axis is moved with small speeds of rotation, the IPP will work with high step resolution in 1/8th step mode. So we have a smooth running stepper motor, noise emmissions and the influence of resonances are reduced significantly. With increased speed of rotation the step resolution is reduced to 1/4 and 1/2 step mode. At full speed the motor is driven in the powerfull full step mode. The stepp failure indication unit (option) is adapted for this dynamic step width change and will detect any step loss of the system.

- Free usable digital inputs and outputs, and separate inputs for limit switches are integrated in the IPP - you do not need any additional equipment.

- Free exchangable system software by means of a flash EPROM:

  System software and all parameters of the axis are stored in a flash EPROM of the IPP. This flash EPROM may be reprogrammed via the serial interface without opening the device. Adaptions to customer specific applications can be made without changing any internal component.

- Optional extension boards

  - A plug-in step failure indication (SFI) unit monitors the stepper motor motion monitored by means of an incremental encoder. The SFI unit will detect any step loss of the system.
  - An extension board with an analog/digital converter enables the evaluation of an analog voltage.
  - An extension board with additional digital inputs and outputs enlarges the number of possible positions to be used in a stand alone application at a PLC up to 256 different locations.

2 Technical Data

2.1 Technical Data of the IPP's Stepper Motor Power Stage

The Intelligent Powerpacks IPP 92-70 or IPP 172-140 contain a high efficient power stage to drive stepper motors. The power stage is built with Power-MOSFET devices, it is designed to drive two phase stepper motors in bipolar operation mode. Stepper motors with 4, 6 or 8 wire connection can be used.

The peak current in the IPP 92-70 is 9 A_{peak}, the effective motor current is then 6.3 A_{eff}. The voltage used at the motor terminals is 70 V_{DC}.

The peak current in the IPP 172-140 is 17 A_{peak}, the effective motor current is then 12 A_{eff}. The voltage used at the motor terminals is 140 V_{DC}.

The main characteristics of the power stages in the IPP 92-70 and 172-140 are:

- Coppered current control scheme according to the SYNCHROCHOP principle as patented by Phytron. With the SYNCHROCOP mode, in medium ranges of speed of rotation the stepper motor will be operated most efficiently.
- Operating current, stop current and boost current (boost means only increasing of motor current during acceleration and deceleration phases) are idividually selectable in 15 steps.
- The step resolution of the power stage is dynamically adapted to the speed of rotation:
  Starting with 1/8th step mode (that means 1600 steps per turn[1]) at low speeds, the system will switch to 1/4th and 1/2 step mode up to the powerfull full step mode depending on the speed of rotation.
- As supply the device needs only one power connection, all auxiliary voltages for the power stage are generated internally.
- Compact, robust housing for wall mounting
- Optimized heat sink. The "big" IPP 172-140 also contains a temperature controlled fan.

[1] All data concerning step resolution are respect to a motor with 200 full steps per turn.

2.2 Mechanical Dimensions

The following drawing shows the dimensions of both the IPP 92-70 and IPP 172-140:

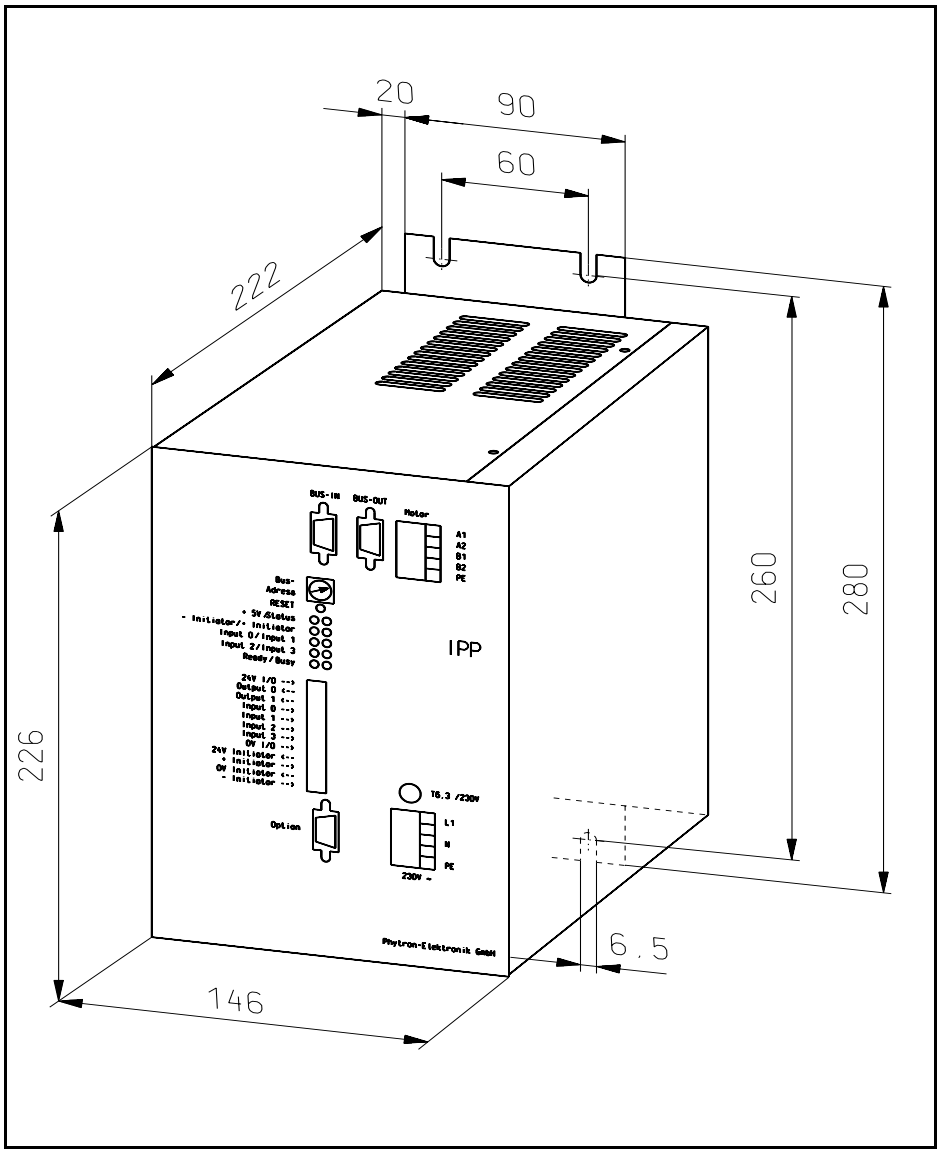


Fig.: 1 Dimensions

2.3 Interface Connectors, Diagnostic Displays

2.3.1 Front Plate Layout

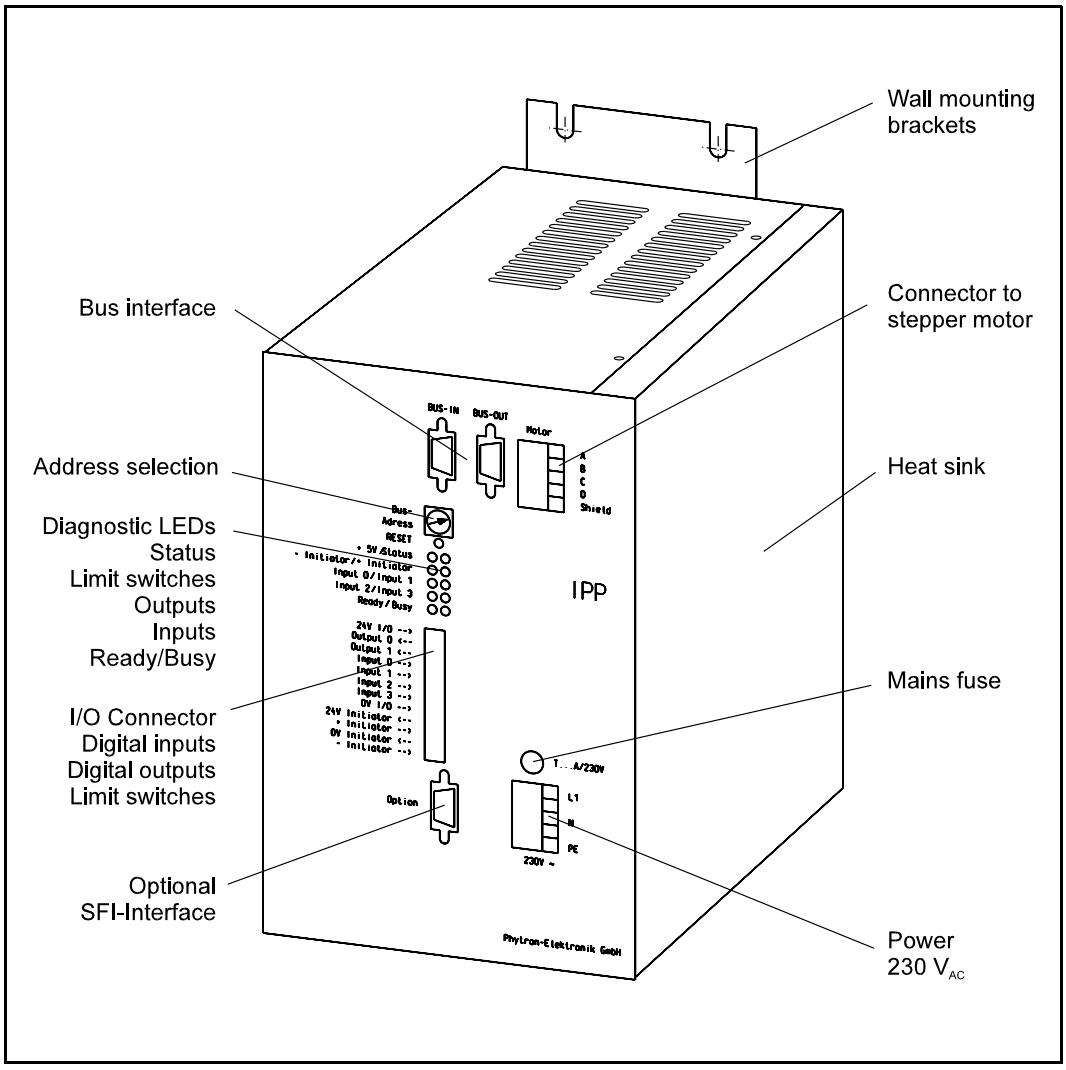


Fig. 2: Front plate layout

At the IPPs front plate, the following components are located:

- Fixed screw type terminals for mains connection
- A mains fuse accessible without opening the device.
- Fixed screw type terminals for the motor connection
- Two type D-Sub connectors called BUS-IN and BUS-OUT for interfaces according to specifications RS-232 or RS-485
- Rotary switch to select the device's bus address
- Light emitting diodes to display diagnostic information
- Screw type-plug in connectors for 4 digital inputs and two digital outputs (nominal voltage 24 V_{DC},

PLC-compatible)

- Screw-type plug-in terminals to evaluate two limit switch signals. Please use limit switches of type PNP-opener as reference and/or emergency stop indicators.

2.3.2 Mains Connector

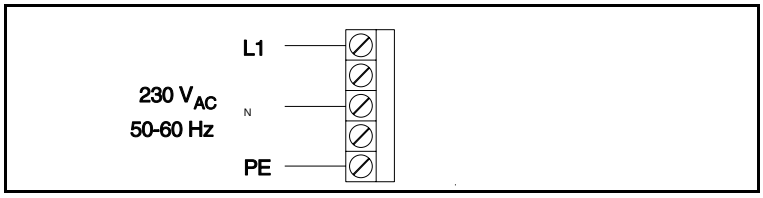


Fig. 3: Mains Connector 230 V_{AC}

Connector type: Phoenix screw-type print clamps DMKDS 2,5

Above the power line connector the mains fuse is loacted. The mains fuses ratings are:

For the IPP 92-70: T3,15 A with 230 V_{AC}

For the IPP 172-140: T 6,3 A with 230 V_{AC}

2.3.3 Motor Connector

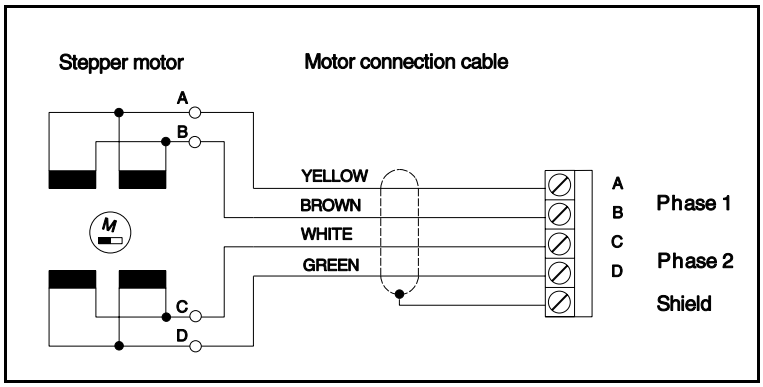


Fig. 4: Stepper Motor Connector

Connector type:

Phoenix print clamps DMKDS 2,5

2.3.4 Bus Interface

Both bus interface connectors called BUS IN and BUS OUT are 9pin type D-Sub connectors (according to DIN 41652) for connecting the IPP to a PC or some other master controller.

The BUS IN holds the RS-232 and the RS-485 interface signals, the BUS OUT connector is used only for the RS-485 field bus connection.

Note: • Use only one of both interface types at the same time.
• You must not have RS 232 signals active while using the RS 485 interface and vice versa!

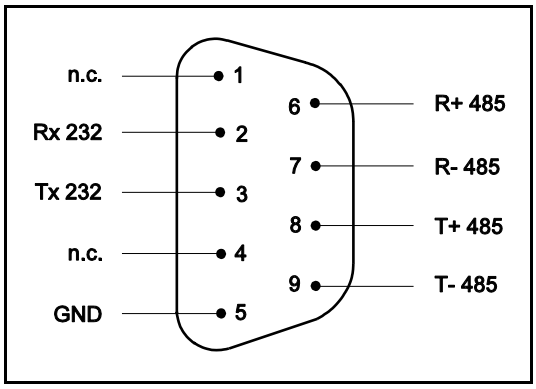


Fig. 5: 9-pole D-SUB Connector BUS-IN (male)

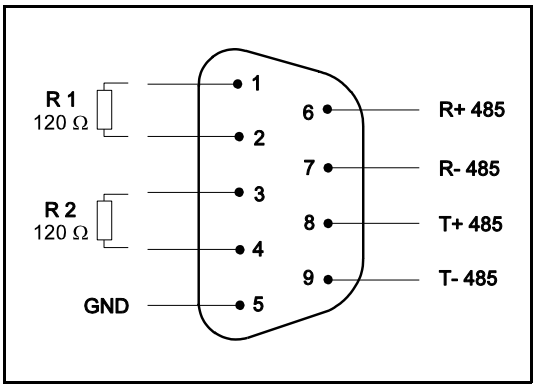


Fig. 6: 9-pole D-SUB connector BUS-OUT (female)

2.3.4.1 RS 232 Interface

If used with an RS-232 connection, only the BUS IN connector is used. In this configuration, one IPP can be connected to one COM-port of a PC. This is the normal way to configure the IPP by means of the software IPCOMM. Please refer to the cabling information below to see how the cable should be made.

Notes: • In the RS-232 operation mode the BUS OUT connector must not be used for any purpose!
• Please select the appropriate device address if using the RS232 mode!

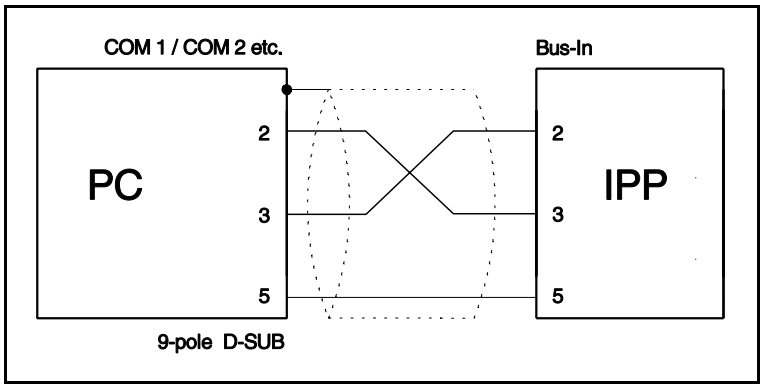


Fig. 7: Interface Cabel for RS 232 Operation (9-pole D-SUB at PC)

Warning:
- The RS 232 connections pin 2 and pin 3 are not to be chained!
- With RS 232 operation mode no termination is necessary.

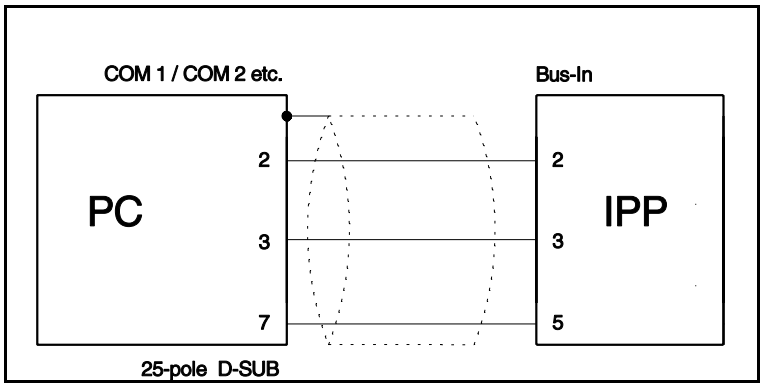


Fig. 8: Interface Cabel for RS 232 Operation (25-pole D-SUB at PC)

2.3.4.2 RS 485-Interface

If operating more than one IPP device at one serial interface of your master controller, the RS 485 field bus is the best choice. The IPP has a 4 wire RS 485 interface, optically isolated from any other circuit in the device. With its simple, but efficient and fault tolerant protocol the RS 485 field bus defined by Phytron fits best for applications in industrial environments. Please refer to the separate manual “IPP Protocol Specification”, if you want to implement complex systems with several axes by yourself.

Note: If you use the RS 485 field bus, please select unique axis IDs (addresses) for each IPP device.

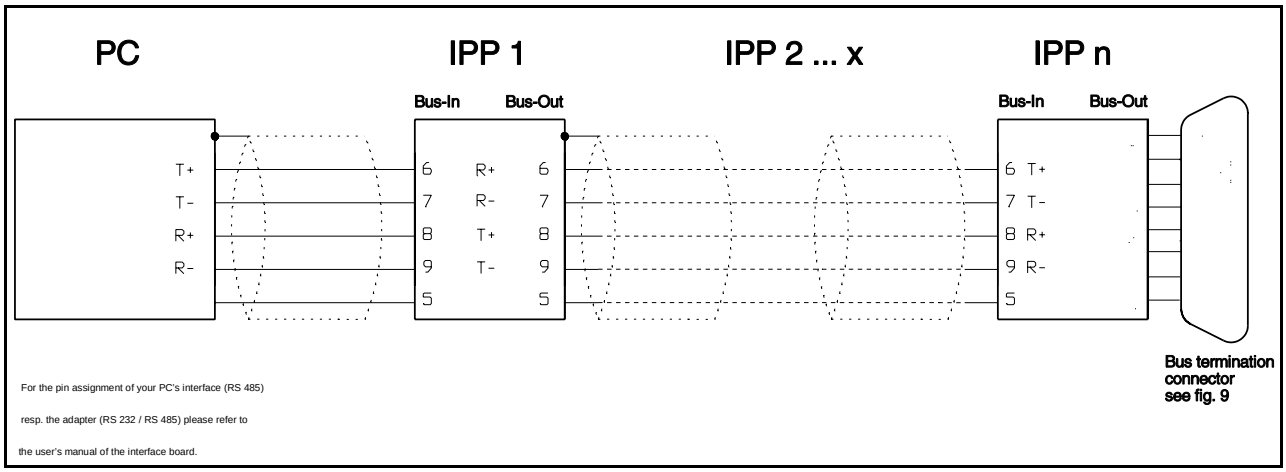


Fig. 9: Interface Cable for RS 485 Operation

To have well defined signal levels on the RS 485 bus, is is strongly recommended to have a bus termination at both ends of the bus. Normally, at one end of the bus the master controller will do this. At the distant end of the bus please use the integrated bus termination resistor network at the very last IPP in the chain. Each IPP contains bus terminator resistors, and by applying a dummy D-Sub connector with 4 little bridges in it these terminaton resistors are switched onto the bus (refer to drawing 9).

By doing this the RS 485 bus will have well defined levels in its idle state.

The bus termination dummy D-sub connector (male 9 pin) is to be plugged onto the BUS OUT connector.

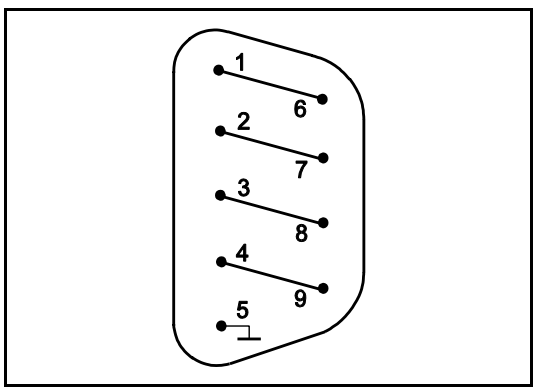


Fig. 10: Bus Termination Connector for RS 485

Recommendations for correct cabling:

1. To reduce the influence of noisy environments only shielded cables and connector housings should be used. Please check whether the connector housings are well both fixed to the IPP and the PC to have a very good contact from the cable shield via the housing to the device.

2. The cable shield should be connected to the connector housing and through it to the housing of the IPP and/or PC on both sides. This will reduce noise sensibility, if the transition resistance of this shield connection is low enough and the shield is connected with a large cross

section (NOT by a single wire!).

2.3.5 Limit Switch Connector

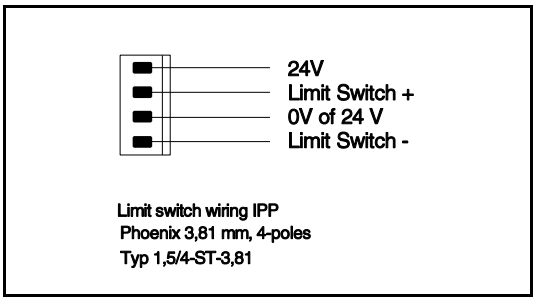


Fig. 11: Limit Switch Connector

Connector type:

Phoenix MC 1,5/4-ST-3,81, screw-type plug-in terminal.
Up to two limit switches of type PNP-opener may be connected (nominal voltage 24 V_{DC}).

Auxiliary power supply:
The IPP can deliver auxiliary power (24 V_{DC} rated at 100 mA) for supplying the limit switches.

If any of the limit switches is missing - e.g. if you do not need this supervisory function in your application- you have to tie this input to 24 V to prevent the IPP from detecting an emergency stop condition.

2.3.6 Digital Inputs and Outputs

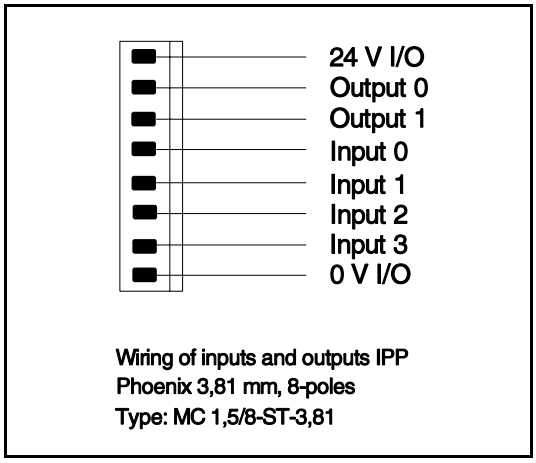


Fig. 12: Digital I/O Connector

Connector type: Phoenix MC1,5/8-ST-3,81

Screw type plug in connectors for four digital inputs and two digital outputs.

Digital inputs:

Nominal voltage:	24 V_{DC}
Threshold voltage (0-->1):	about 5 V
Nominal current at 24 V:	about 10 mA
Maximum voltage:	30 V_{DC}
Isolation to power line, serial interface, motor voltage:	500 V

Digital outputs:

Nominal voltage:	24 V_{DC}
Nominal current:	< 50 mA
Switcher type:	open emitter with optocoupler (darlington scheme)
Supply:	to be fed in externally
Overload protection:	thermically

2.3.7 Voltage Levels

The following table shows the signal levels and functions associated with the single input/output signals. For reference about the meaning of each signal (using the IPP in a stand-alone application combined with a PLC) please refer to chapter ??.:

Signal	Function	Level	Comment
Serial Interfaces			
RS 232 BUS IN	Serial interface according to RS232 or RS485 (selected by wiring)	a, Standard RS232 type interface with +/-9 V voltage levels b, Standard RS485 type interface for 4 wire connection, differential signal driven from 5 V auxiliary voltage	a, Do not use BUS OUT with RS232 interface! b, RS 485 signals are connected in parallel to BUS IN connector (passive, no repeater) c, At the distant end of the bus use a dummy connector with bridges for bus termination.
BUS OUT	a, Serial interface RS 485 b, Resistor network for bus termination	RS485 (4-Draht), 5 V differential	
Digital Inputs and Outputs			
24 V I/O	24 V_{DC} to be fed in as auxiliary power for the digital outputs	Input for auxiliary voltage to drive the digital outputs; nominal voltage 24 V_{DC}	Outputs 0 and 1 are isolated against all other inputs and outputs.
Output 0	Digital output 0	Switches auxiliary power **24 V I/O** to this output, if active.	Each output can deliver up to 50 mA and is short circuit protected.
Output 1	Digital output 1		
Input 0	Digital Input 0	PLC-compatible input, nominal voltage 24 V_{DC} respect to the terminal **0 V I/O**.	Threshold voltage is about 5 V_{DC}.
Input 1	Digital Input 1		
Input 2	Digital Input 2		
Input 3	Digital input 3		
0V I/O	Common ground for digital Inputs.		
Limit Switch Inputs			
24 V Limit Switch	Auxiliary power source to drive limit switches without external power supplies.	nominal Voltage 24 V, short cicuit protected, max. current about 200 mA.	
+ Limit Switch	Input for limit switch at positive end of axis range		
0 V Limit Switch	Common ground for limit switch inputs	also ground terminal for the auxiliary limit switch power supply	
- Limit Switch	Input for limit switch at negative end of axis range		

Signal	Function	Level	Comment
Option	D-Sub type connector, 9pin, female, for optional stepp failure indicator (SFI) module.		
Motor Terminals			
A	Stepper Motor Phase 1	IPP 92-70: 70 V isolated IPP172-140: 140 V isolated	The cable cross section has to be large enough to handle the maximum current continuously: Current limits are: IPP 92-70: 9 A IPP 172-140: 17 A
B	Stepper Motor Phase 1		
C	Stepper Motor Phase 2		
D	Stepper Motor Phase 2		
Shield	Cable shield, internally connected with PE	Must not be used as PE (protective earth) for the motor itself!	
Power Line Input			
L1	mains phase		The cross section has to be large enough to handle the maximum power line current: IPP 92-70: 6 A IPP 172-140: 10 A
N	mains neutral wire		
PE	protective earth		

2.3.8 Diagnostic LEDs

The IPP devices contain the following light emitting diodes as diagnostic elements:

Name	Function	Comment
+ 5V	internal power supply	The internal auxiliary voltage to supply the axis controller is active, the power line is active.
Status	Activity LED for the seriel interface	a, During each access to the IPP via ther serial link the LED will flash for a very short period. b, After power up the LED will go on for about 2 secs after the power on self test has been done. c, If an internal error is detected, the LED will go on and stay on until the error is reset by software or reset. Possible errors are: • Limit switch error (both limit switches active the same time) • Overtemperature detected (heat sink temperature reaches 85°C) • Motor current limit exceeded (short circuit detected)
Limit Switch +	+ Limit switch is damped	Control signal for limit switch diagnostics
Limit Switch -	- Limit switch is damped	Control signal for limit switch diagnostics
Input 0	Digital input 0	The LEDs are used to show the current state of the 4 digital inputs and the 2 digital outputs. Note: The LEDs are only active if the power is on. The LEDs cannot be driven by external voltages alone.
Input 1	Digital input 1	
Input 2	Digital input 2	
Input 3	Digital input 3	
READY	Digital output 0	
BUSY	Digital output 1	

3 Putting an IPP into Operation

3.1 Things You Need for the First Test of an IPP

If you are going to perform the first tests with an IPP, you should have available the following devices:

- A PC with the configuration software IPCOMM already installed. Please refer to chapter ?? how to install the software IPCOMM.
- A data cable to connect the IPP to your PC. We recommend to do the first steps with the RS 232 connection and not to start with a RS 485 field bus system. Please refer to chapter ?? how such a cable should be wired.
- An IPP with a stepper motor and the appropriate cable to connect the motor to the IPP.
- A cable to connect the IPP with the power line, and eventually the cabling to connect the limit switches. If you do not have limit switched available, please simulate limit switches by making 2 small bridges at the limit switch input connector: Simply connect the +24 V_{DC} auxiliary power supply with both the + limit switch and - limit switch input.

  Note: Please remind that with this bridge no range checking and no emergency stop feature are possible!
- To work with the configuation program IPCOMM you should have some experiences with the operating system WINDOWS ®. Please refer to your PCs manual how to work with this operating system.

3.2 First Steps with an IPP

- Connect the stepper motor with the IPP.
- Connect the IPP with the power line. Do not forget to use the PE wire, too.
- Connect the limit switches with the IPP, or install bridges to operate the IPP without limit switches.
- Connect your PC's RS 232 COM-port with the BUS IN connector of the IPP. Be aware, that the pinning of standard COM-port differs from 9pin to 25pin versions!
- Check the axis ID (address) of the IPP you use. Any change of this rotary switch comes into action only during a reset or power up sequence. If the address is not set up in a consistent way, no communication is possible.
- Turn on the power of the IPP: The red LED named **Status** should flash for about 2 secs after turning power on.
- Now start the software IPCOMM on your PC. Select the appropriate communication port (in the field **interface** at the left side of the screen). Then activate the axis number you want to use. Please be careful, the ID you select in the field **IPP axes (existing)** and at the rotary switch at the IPPs front panel have to be the same! You have to select an axis in the left group on the screen, and then you can open the axis window either with **Communication** via the main menu or directly by clicking with your mouse onto the button with the axis' number.

Note: If you select “None” as setting for the communications port, IPCOMM will not try to access the IPP. Due to the fact, that all data displayed on the screen has its origin in the IPP’s memory, IPCOMM will show only some functions if used without an IPP connected to it.

- If the window appears without any error message, IPCOMM has established an online connection to the IPP. Please press now the **STATUS** button to see the internal state of the IPP. Please close this dialog box again.

- As first operation, you can now select an appropriate motor current: Press the button **CURRENT** and select the correct value for stop current, operating current and boost current.

  Note: With earlier versions of IPP hard- and software, the current values are shown only in current steps. If this happens, close the current window and re-open it again. If there still current steps and not amps are displayed, you should update the firmware of your IPP.

- As next, please go back to the main axis window (titled AXIS x). Now select the **motion command** move relatively. Enter some distance in the field named .. to / .. by. And now press the start button. Your motor should run now!

- If something goes wrong, please refer to the chapter problem solving.??

4 Test and Configuration of an IPP with the Software IPCOMM

4.1 Overview

With every IPP Phytron delivers a special software IPCOMM **free of cost**. IPCOMM runs under Windows 3.1 and Windows 95. With IPCOMM you can easily

- test the functionality of the IPP,
- test the dynamics of your drive to evaluate the best axis parameters,
- set up all IPP internal parameters and
- define fixed program sequences for stand alone operation controlled by a PLC device.

IPCOMM also helps you to make backup copies, documentations and complete images of IPP internal data for archiving purposes.

4.2 Hardware Requirements to Use IPCOMM

To use IPCOMM on your PC the following hardware requirements should be fulfilled:

1. You need at least an 80386 based PC with WINDOWS® Version 3.1 or above.
2. At least 4 MB of working memory (RAM).
3. Free hard disk space of about 5 MB
4. Two serial interfaces (mouse and IPP) or one serial interface if you use a bus mouse.

Note: Most of the IPCOMM functions can also be started by means of keyboard input, but not all of them. So a mouse is necessary.

Note: IPCOMM can only be used, if an IPP is connectoed to your PC. IPCOMM does not support a demonstration mode, because IPCOMM uses all data as it is read out of an IPP device.

4.3 Installation of IPCOMM

IPCOMM is delivered on (a) floppy disk(s). On the first of these you will find files like LIESMICH.TXT (German) or README.TXT (English). These files contain additional information written after the edition of this manual. Please read these files carefully, because the information contained therein supersedes this manual.

To save floppy space, the IPCOMM files are delivered in a compressed format. During the installation procedure these files will be expanded automatically.

4.3.1 Installation of IPCOMM under Windows 3.x

To install IPCOMM, please proceed as following:

1. If you have already an elder IPCOMM software installed in your PC, please make a backup copy of this version, and all files edited with this version!

2. Start your PC and start Windows.

3. Insert the floppy disk named “IPCOMM disk 1 of ...” into your floppy disk drive.

4. In the Windows program manager now select ***File | Execute***. Enter e.g. **A:\SETUP** in the command line or use the button ***Search*** to look for the setup file SETUP.EXE on the floppy disk.

5. Start the SETUP program.

6. SETUP will ask you for a directory for the IPCOMM program files, and another directory for the project files. If your IPCOMM copy just supports native language support, you are asked to define the language to be used in your installation.

   Note: If you are using Windows 3.11, a very special problem may arise and SETUP eventually shows up a message like “Temporärdatei konnte nicht angelegt werden” or “Temporary file could not be created”. Refer to the hints below to circumvent this problem.

7. Now SETUP will copy the program files into the destination directories.

8. With IPCOMM we deliver some drivers not included in the standard Windows 3.1 disk set. SETUP will copy these files in your windows system directory (normally C:\WINDOW\SYSTEM). SETUP will issue a warning message, if you have these drivers already in your system directory. Please select IGNORE if you don’t want to update these drivers.

4.3.2 Installation of IPCOMM under Windows 95

To install IPCOMM under Windows 95, please proceed as following:

1. If an older version of IPCOMM is already installed on your PC, please make a backup copy of this version and all files edited with it!

2. Start **System Control** and there **Software** and then press the button **Install**. Now enter in the command line the string **A:\SETUP.EXE** (if your floppy disk drive is named A:).

3. Now follow the instructions on the screen. Please refer to “Installation of IPCOMM under Windows 3.1" for further hints, because the rest of the installation procedure is the same for both Windows versions.

4.3.3 Registration of Software Modules (Tools)

IPCOMM has a very flexible, open structure: One main program, IPCOMM.EXE, can be adapted to different applications by adding library modules, DLLs. In these DLLs tools are implemented which give you the access to different internal parts in the IPP.

To use these tools, you have to register it, so IPCOMM knows what tools you have on your PC. To register these tools, please

1. Start your PC, start Windows and start IPCOMM.

2. Use ***Tools / Register PLC-Modules*** to register DLL modules to handle PLC functions within the IPP. Two

different types are available: PSNORMAL.DLL handles the basic PLC-interface (4 digital inputs, 2 digital outputs) while PSEA.DLL handles the extended PLC functionality if your IPP has the I/O extension board.

3. Use ***Tools / Register Ramp-Modules*** to register DLL modules to handle different ramp generators within the IPP. Up to now only linear ramps are possible, so please register PRLINEAR.DLL.

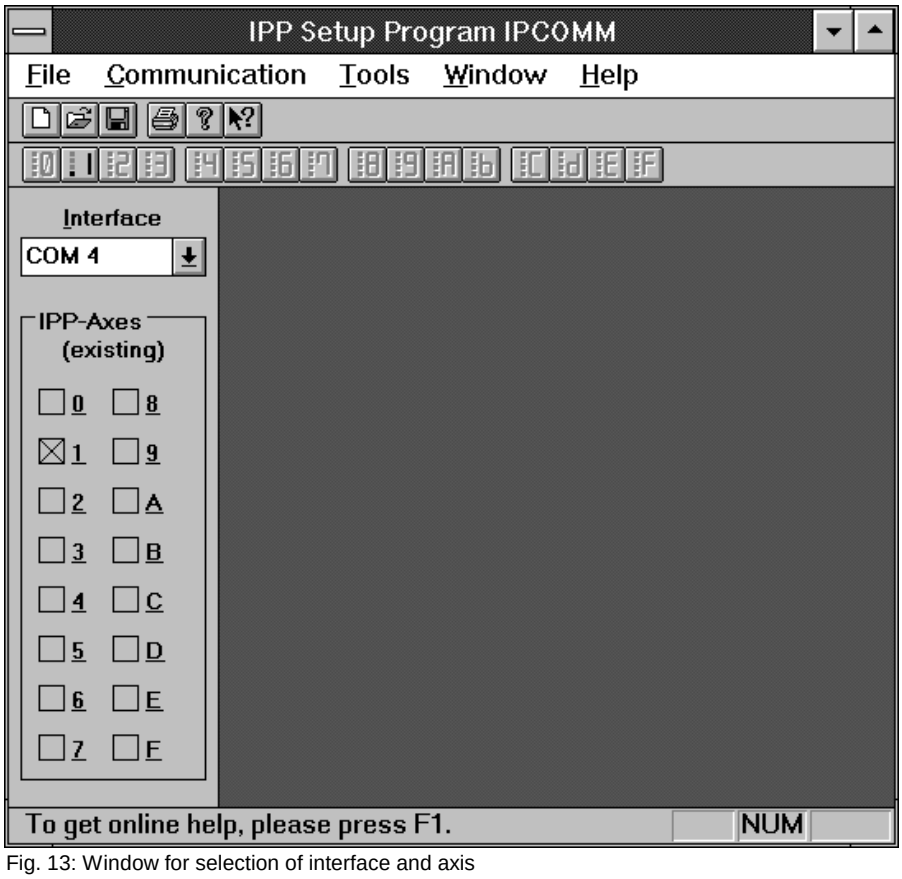
4. Use ***Tools / Register IPP-Options*** to register DLL modules to optional extension boards delivered with the IPP. Available up to now are POAD.DLL (analog/digital converter extension) and POEA.DLL (digital I/Os).

4.4 First Test of IPCOMM with an IPP

After installing IPCOMM and connecting PC and IPP by cable you can test the system. Start IPCOMM and select two other parameters:

- Please select the serial communication port you use (see the field ***Interface***).

- Please select the axis you want to communicate with in the field ***IPP-axes (existing)***. Here you tell IPCOMM what axes exist in your application. Be sure that this address you select is the same you have selected at your IPP.



Note:

Fig. 13: Window for selection of interface and axis

After changing the IPP's address you have to reset the IPP manually, because the address selection is read out only once during power up.

- Now you can open the axis window by using ***Communication*** or simply by clicking onto the axis symbol in the button bar located below the menu bar.

- By clicking onto the ***Status*** button you send a status request to the IPP, and a status window will be shown. If there is any problem with the communication, a message appears, that there has been a timeout during the communication. In this case please check the cabling. Refer to section trouble shooting for further hints to solve the problem.

5 Overview over the IPCOMM Functions

IPCOMM is a very easy to use software to configure and set up IPP devices.

In the next chapter 6 titled “Reference list of the IPCOMM Functions” you will find explanations in detail. In this chapter you will get a fast overview over the functions in the different windows and dialog boxes.

After selecting the interface port and axis ID in the main window, you can activate the main axis window for the IPP you use. The main axis window is labelled “AXIS x” in its title bar. For every axis you work with simultaneously such a window exists. Up to 16 axes can be controlled with IPCOMM the same time.

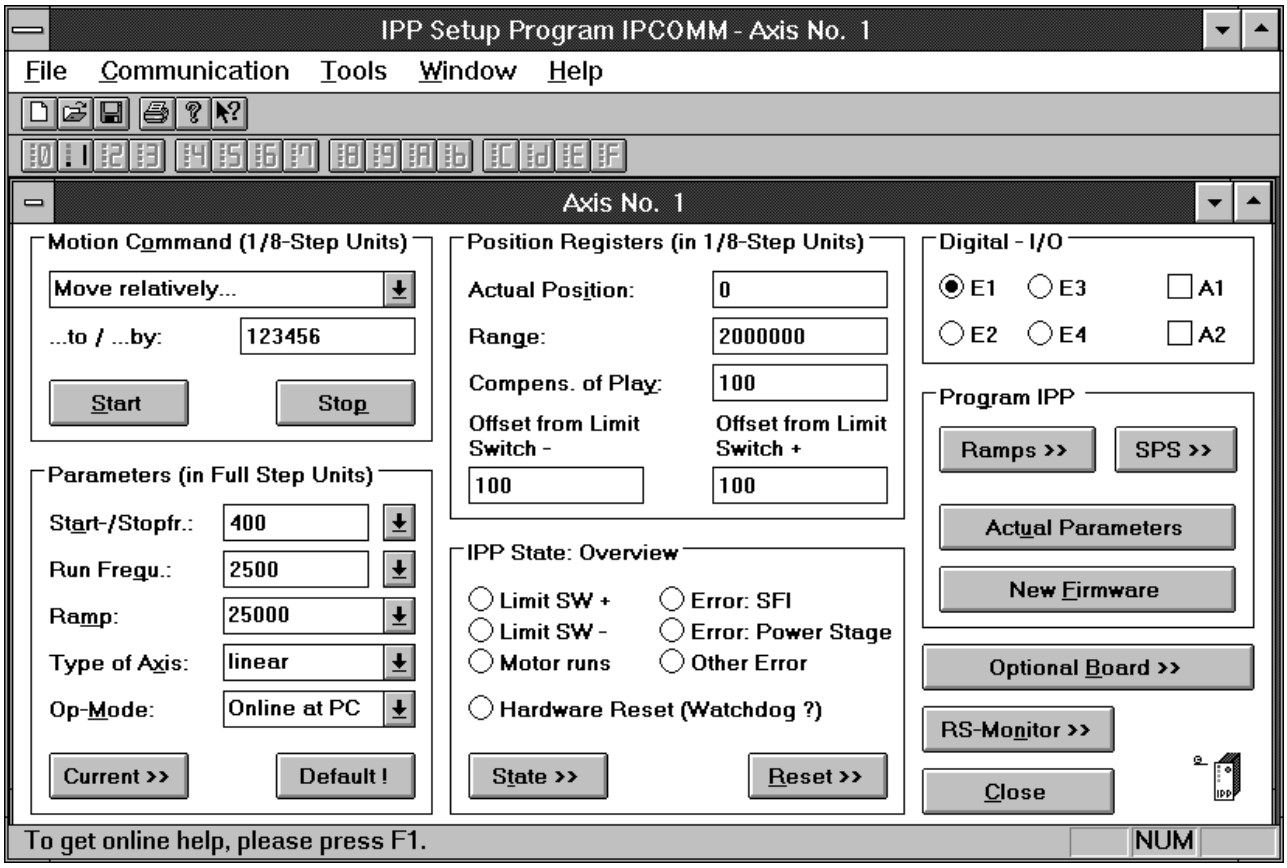


Fig. 14: Main axis window - overwiew

Note: IPCOMM will save the settings for interface port and axis ID if you wish it. So if you start IPCOMM later again with another axis ID used with your IPP, it is most probably that you get the message “Timeout with the serial interface”. Please abort this message with CANCEL several times, and close the axis window with CLOSE to get back to the empty main window with the interface and address selections, and select then the correct parameters.

In this main window you see several groups of fields and some buttons to open sub windows with more detailed information. Most of the elements in this window may be accessed with keyboard shortcuts (e.g. Alt-S causes the same action as pressing the Start button).

In the following list the single groups are explained:

Motion command

Here you can select one of the available basic motion commands like “go relative” or “go to absolute position”. In an edit field you may enter the distance you want the motor to go. Please remind, that this distance is always given in 1/8 full step increments, because the axis will position to this value (frequencies on the other hand always are respected to full steps). With START and STOP you can start the motion command and also abort it.

Parameters

Here the basic axis parameters like frequencies, ramps and operation modes are defined. Please note that ramps are free programmable, but not automatically. In parameters you can only select one out of the predefined acceleration ramps. How these ramps can be changed and redefined, you will see later.

With a special button in this group you can open the current setup dialog box.

Warning: Set up the correct motor current before giving any motion command! Take care that the motor current is not too high. Due to the wide range of possible current values with the IPP your motor may be damaged!

Also note that any tests with the IPP from this main axis window will work only if the operation mode is set to PC-online.

You can reset the IPP to its default values by pressing the **Default!** button.

Position Registers

Here you see the actual position counter of the axis connected. In the other field you can enter data like offset values used during reference runs, some compensation of gear play and a range limit.

IPP State: Overview

This group shows the values of the current state bits as read out of the IPP. The bits shown are accumulated from single error bits. All but **motor runs** will cause the IPP to abort any motion command being in progress. The error bits are latched for your convenience, and you have to use **Reset>>** to reset any error condition. **State>>** will show up another window with detailed error informations.

Digital - I/O

This group of elements shows the current state of the four digital inputs and the two outputs of the IPP. The current state of the outputs of the IPP may be remotely controlled from here.

Program IPP

Here some buttons are shown which open special sub windows e.g. for ramp definition and for the definition of PLC sequences (see chapter 6 for the details).

Note: Herr you will find the button **Actual Parameters>>**. All changes you perform in the main axis window will become stay permanent in the IPP only if you select this button at last. If you don’t do this, changes in current, frequencies and so on will be lost during the next power up sequence of the IPP.

Optional Board and RS-Monitor open sub windows which are explained in detail in chapter 6.

6 Reference List of the IPCOMM Functions

6.1 Parameter Definitions and the Online Operation

For each axis IPCOMM has a window for setting the motion parameters:

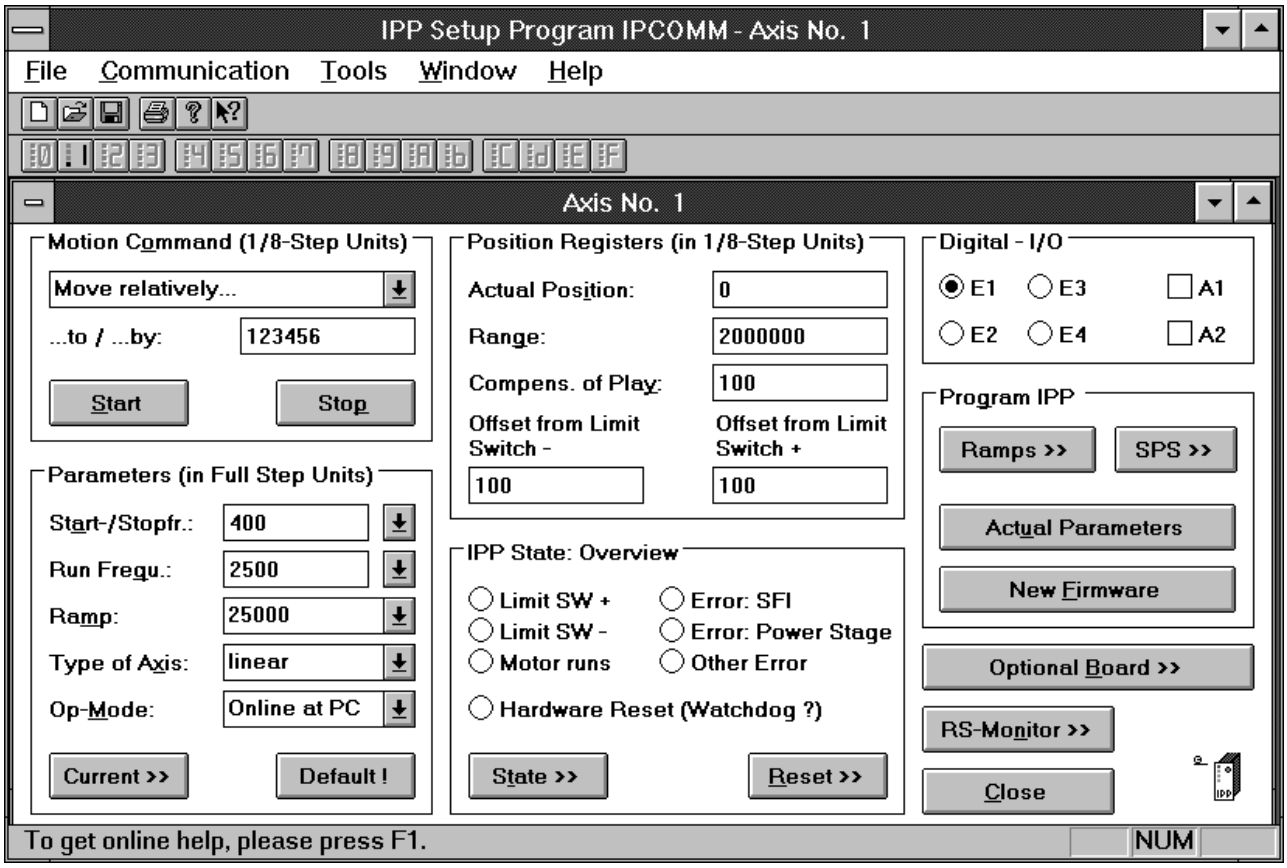


Fig. 15: Main axis window with edit field for the motion parameters

In the following each element of this main axis window is explained in detail:

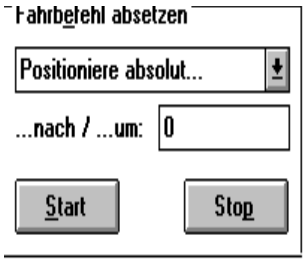


Fig. 16: Field: Motion Command

Field: Motion command

Here you can select a motion command by means of a list box. Depending on the type of command you select the input field ...to / ...by disappears. For the motion commands **Go relative** and **Go absolute** you have to enter the destination position/distance the axis should move.

Note: All motion commands are defined to be with 1/8 full step resolution.

Other possible commands are **Initialization with + Limit Switch** and the same for the limit switch at the negative end of the axis. Initialization means that the motor is positioned with normal speed toward the limit switch, if this is not damped (free). If the limit switch has been found, the axis will go back with slow speed (start/stop frequency) until the transition damped/not damped has been found. If a limit switch is defective and the initialization cannot be completed the axis would run forever. Therefore the axis will stop automatically if the allowed range has been exceeded (see **Range** on page ?).

With the buttons **START** and **STOP** you initiate the motion command's execution and abort the command. The **STOP** command causes the axis to slow down with the defined acceleration/deceleration ramp.

Field: Parameters

In this group of input fields the basic parameters of the axis can be defined. Please consider that the frequencies shown here are in full step units.

Offset Frequency:
This is the frequency whith which the stepper motor will perform its very first step. The value you enter here should be adapted to you mechanical system. Typical values are in the range of 100 Hz up to about 800 Hz, depending on the motor size and the moment of inertia of the system. IPCOMM will accept start/stop frequencies in the range from 50 Hz to 1000 Hz. If the start stop frequency is defined to be higher than the run frequency, the smaller of both is used permanently.

Run frequency:
This is the maxmimum frequency which will be reached if the motor is accelerated to its full speed. The physical limit of the IPP's hardware is 10,000 Hz full step, this means 80,000 Hz in 1/8 step mode[2].

Note: As usual, this frequency is reached only if the axis is moved some minimum distance, depending on the acceleration ramp slope.

If the frequency selected is below 50 Hz, the IPP will move the motor only with start/stop frequency. There will be no ramp generated in this case.

Ramp:
Here a listbox is displayed with the slopes of the already programmed ramp data sets in the IPP. The ramp slopes are given in Hz/sec in full step units. Up to now the IPP can drive the motor only with linearily increasing or decreasing ramps, and only equal ramp slopes for acceleration and deceleration phases can be set.

If the IPP's ramp tables have been deleted, or if the communication between PC and IPP had a timeout condition, the **Ramp** listbox remains empty.

Note: The ramp values shown here are stored in a table within the IPP's flash memory. You can modify this acceleration ramps in a wide range (refer to chapter ? below). But you can use only ramp tables which have been programmed before. The IPP cannot calculate ramp tables in realtime.

[2] The basic hardware of the IPP may be operated up to 18,000 Hz full step if equipped with a faster microcontroller. Please ask for availability if your application needs such high speeds.

Type of Axis:

Here the distinction is made between a linear and a rotational axis. With rotational axes, the limit switch is used only during the reference run to define a mechanical zero position. During normal operation the limit switch is neglected, and no emergency top sequence initiated if the limit switch becomes active (damped).

With a linear axis, both limit switch inputs are evaluated permanently. If the axis is moving toward positive direction and the + limit switch becomes active (damped), the axis controller will decelerate the axis immediately and stop the motor. The same happens, if the - limit switch becomes active while moving in the negative direction.

Note: After a limit switch has become active the initiator error bit has to be reset manually.

Operation Mode:

There are three different operation modes available (depending on the hardware configuration of the IPP):

PC-Online: This mode is used to access the IPP via its serial interface. The digital inputs are ignored in this case, and no digital output is used.

PLC-mode: In this operation mode the digital inputs and output are used together e.g. with a PLC controller. The PLC is the master, the IPP the slave. With the first start signal received from the PLC the axis will start.

Option-mode: This operation mode is used if a extension board has been built into the IPP. The auxiliary functions of the extension board can only be activated if the operation mode “option-mode” has been selected.

Current:

This button activates a window with input fields for the stop current, the operating current and the boost current (boost here means to increase the operation current only during the acceleration and deceleration phase of a motion). Normally IPCOMM will read the possible current values out of the IPP device connected. During the first factory set up of the IPP it will learn its current values, and so you can see the current values in readable form, e.g. like 3.7 Amps.

Note: If instead of current values IPCOMM only shows *current steps* without the detailed current value, you should update your IPP’s software with some new firmware release.

In the current window also the recovery time can be set. With recovery time some additional delay is set. The system will wait for this time after the very last step has been performed. The motor can slow down its oscillation around the destination position. If you attempt to start the motor once again immediately after the last step there may be step losses. A recovery time of about 20 msecs is the best choice for most applications.

Note: Please remind that any modification of the motor currents will be lost if you do not store them permanently into the remanent memory. Use the button “Actual Parameters” to do this, otherwise any changes will be superseded by the default values during the next power on cycle.

For both devices IPP 92-70 and IPP 172-140 the following table shows the default current steps:

	IPP 92-70	IPP 172-140
Current Step	Current [A_{eff}]	Current [A_{eff}]
0[1]	0	0
1	0.35	0.65
2	0.81	1.53
3	1.2	2.2
4	1.5	2.9
5	2.0	3.7
6	2.4	4.5
7	2.8	5.3
8	3.3	6.1
9	3.8	7.1
10	4.2	8.0
11	4.6	8.7
12	5.1	9.6
13	5.5	10.4
14	5.9	11.2
15	6.3	11.9

Notes to the current table:

- Do not set the run current to 0.
- The stop current may be set to 0, this means “motor is de-energized”.
- If you select the boost current to be at level 0, the current used during acceleration and deceleration will be the normal run current.

Default!:

Pressing this button will set all internal IPP parameters to default values. You will be asked for a confirmation that this should be done.

Field: Position Registers

In this area several offset positions and the actual axis position are displayed.

Note: Please remind that all positions refer to 1/8 step resolution.

Actual Position:

This field shows the actual axis position (in 1/8 full steps). During an initialzation cycle (reference run using a limit switch) this position is set to 0 automatically.

You can set the position counter to any value you need by simply entering new data into this edit field. If the cursor leaves the edit field, the new value will be transferred to the IPP.

Range:

Here you define a maximum distance limit the axis may go. This value is valid for the following operations:

- free run toward + limit switch
- free run toward - limit switch
- initialization towards + limit switch

- initialization towards - limit switch.

With these commands the motor might run forever in case of any external fault. To prevent the mechanical system from any damage these motions should be limited by means of this field. If you set the limit value to 0, no range check is performed at all.

Compensation of Play:

This parameter is used to get a more correct final position if you are using a gear by taking into account some play. If compensation of play is set to a value different from zero, the final position during normal positioning sequences (**go to absolute position** and **go relative**) will always be approached from one side of the axis range. The details are shown in the following table:

Examples for compensation of gear play:

Hint: The command **GR** in the following table means *go relative.*

Input in edit field **Compensation of Play**	Effect	Motion after sending a motion command “move towards + limit switch” e.g.: **GR+1000**	Motion after sending a motion command “move towards + limit switch” e.g.: **GR-1000**
< 0 e.g. **-100**	Compensation of play works only if motion is towards - limit switch	1000 steps in positive direction	Axis will go 1100 steps in negative direction, then turn and go back 100 steps in positive direction
0	No compensation of gear play	1000 steps in positive direction	1000 steps in negative direction
> 0 e.g. **+100**	Compensation of play works only if motion is towards + limit switch	Axis will go 1100 steps in positive direction, then turn and go back 100 steps in negative direction	1000 steps in negative direction

Offset from Limit Switch +:
Offset from Limit Switch -:

Here the offset position is defined which will be used immediately after a reference run towards the positive/negative limit switch. After having searched for the edge of the limit switch signal, the IPP will go for this offset value and then set the position counter to 0.

Field: IPP State: Overview

In this area the current state summary is displayed. All but the “motor is running” signal are latched until they are reset manually. Please note that there may be some time lag between the actual event and the display on the screen due to the transmission time of the serial interface.

Limit SW +: Limit switch + direction is damped or missing
Limit SW -: Limit switch + direction is damped or missing

Motor runs: Shows that the motor is currently running (this signal remains until the recovery time elapsed).

Error SFI: A step failure was detected. This item is only valid, if a step failure indicator module (SFI) is installed. Refer to the description of SFI in the appendix.

Error: Power Stage:
There has been any power stage fault (overtemperature, short circuit or power fail).

Other Error: If this error message is displayed there has been for instance an attempt to change the running frequency while the axis is still moving - this is not possible, and therefore the request is denied by the IPP.

Status: With this button a status dialog box is activated. In this dialog box the single error bits of the extended status code are shown. For details please refer to chapter 6.2.

Reset: Used for reset after errors during the test phase.

Field: Digital-I/O

In this area of the main axis window the state of the digital inputs and outputs is shown (only for the basic I/Os but not for the I/Os of the extension board). The input display is passive. The outputs can be set and reset by a simple mouse click.

Button: RS-Monitor

This button activates a dialog box where you can send commands to the IPP by means of the IPP protocol. These tests require a detailed knowledge of the protocol itself. So we recommend the use of this dialog box only to the experienced user.

Warning: The IPP series uses a new defined protocol which is not compatible or even similar to those used by SAM, IXE ot PLS devices!

Button: Actual Parameters

With this button the transfer of all temporarily changed parameters into the remanent memory of the IPP is started. IPCOMM will ask the user for a confirmation.

Note: If this programming cycle is not performed, the IPP will use its original parameters as frequencies, currents and operation modes after the next power up sequence. The ramp information and the PLC program sequences, however, are programmed elsewhere. The selection of the PLC mode is only permanently set if **Actual Parameters** have been programmed!
If you want to program the actual parameters it is good practice to check them after having the IPP switched off and on again after some delay.

Button: new Firmware

Here a complete download of the system's firmware is initiated. The button is deactivated for firmware versions older than V2.xx.

Button: PLC

Here a dialog box appears where the command sequences to be used in the PLC mode can be defined. In the PLC mode, the IPP reacts on digital input signals delivered by a PLC. The PLC is then the master, the IPP the slave, and any serial communication in the PLC mode is only for diagnostic purposes. Please refer to chapter ?? for the details of the PLC mode.

Button: Ramps

Activates a dialog box where the user can inspect, delete, modify and add ramp table entries. Please refer to chapter ?? for the details.

6.2 Status Display

The status dialog box allows to inspect further, more detailed status information. Most of this information is self-explaining.

Hint: In this dialog box the actual status bits are not updated continously.

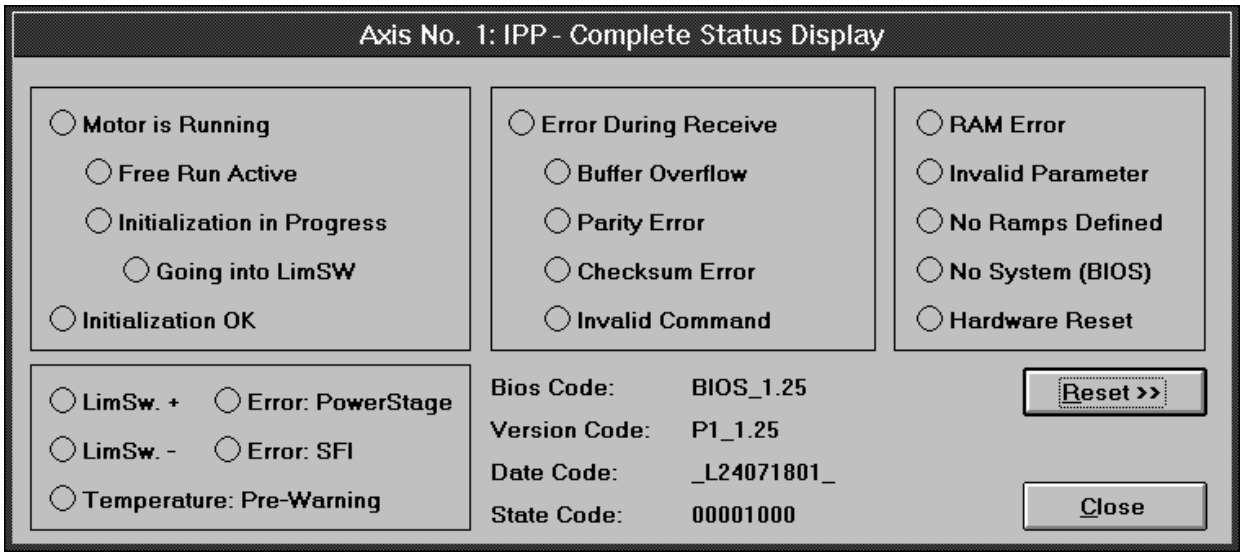


Fig. 17: Status dialog box

6.3 Ramp Programming

The IPP drives the stepper motor using linear acceleration and deceleration ramps. That means, the step frequency is increased or decreased linearily. The ramp data is stored as a table. The table data cannot be modified during the motors operation. With IPCOMM you can program the ramp data once for ever in order to adapt the slope to different applications. These ramp data is the fixed in the flash EPROM until you reprogram it.

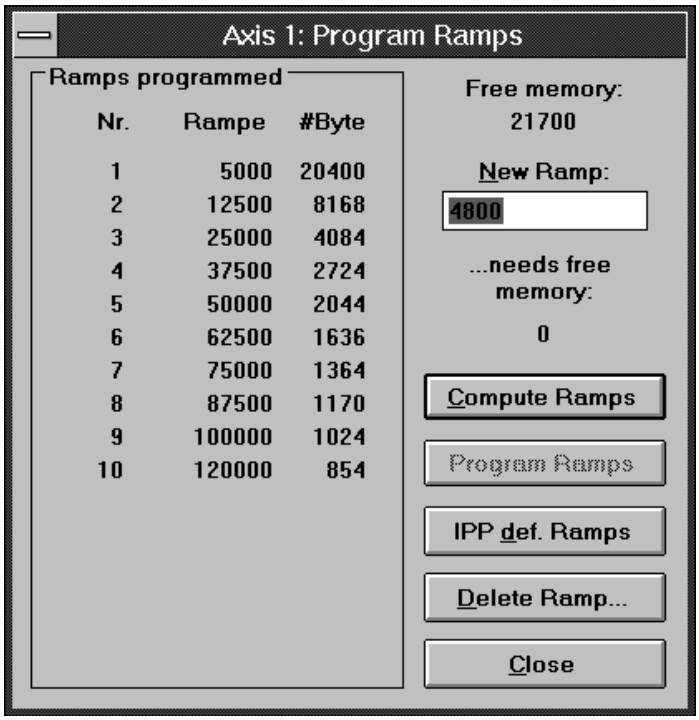


Fig. 18: Dialog box ramp programming

Because the ramp generation in the IPP is based on table data stored in the flash EPROM, ramps may be easily changed by means of IPCOMM.

In the dialog box there are several buttons and display fields available:

Ramps programmed:
Here a list of all currently defined ramps is shown. The entry code in the table, the slope in Hz/s full step and the memory space required are shown.

Free memory: Here the available free memory is displayed. Any new ramp table data must not exceed this space. So if you want to reprogram a flat ramp table (flat = many data items) you probably should make available some memory space with **Delete!**.

New ramp: In this edit field the slope of any new ramp table entry may be entered.

Needs free memory:
Here the amount of memory required by another table entry is shown if you have asked for a test calculation of the new entry with **Compute Ramps**.

Compute Ramps:
With this button you start the computation for the ramp table for the acceleration value you defined in the field **New Ramp**.

IPP int. ramps: Will reset the ramp table of the IPP to the factory default values. First you have to delete the ramp table explicitly by using the **Delete!** button.

Delete!: Will clear the ramp table memory in the IPP.

Program!: With this button the actual programming sequence is initiated in the IPP. All existing ramp data will be deleted, and the ramp memory will be overwritten with the newly defined data. Please note that any interruption of this sequence will leave the IPP's memory in an undefined state.

Note: If you are changing the IPP's ramp table, eventually defined PLC command sequences are deleted, too. This is made because the PLC command sequences contain references to ramp table data.

7 IPP Operation controlled by an PLC

With the axis controller built into the IPP you can easily control a stepper motor by sending commands from a PLC to the IPP by means of digital I/Os. The motion commands are defined by means of the software IPCOMM and then stored in the IPP's internal memory. If the IPP is then configured for PLC-mode, the PLC can start one out of the predefined command sequences.

On oder to have a well defined interface between the PLC and the IPP, you have to use a sequential protocol by using the digital I/Os of the IPP. So the actual state of any basic cycles can be determined at any time.

7.1 Interface Definition in Basic PLC Mode

In the basic PLC mode the 4 digital inputs and the two outputs are used for the communication between PLC and IPP. These I/Os are optically isolated from any other ciruit in the IPP and need an external auxiliary supply of 24 V_{DC}. The 4 digital inputs are used as one STROBE signal as triggering input and 3 program selection bits. So in PLC mode you can perform up to 8 different (3 bits) commands.

The basic command sequence is shown in the following figure:

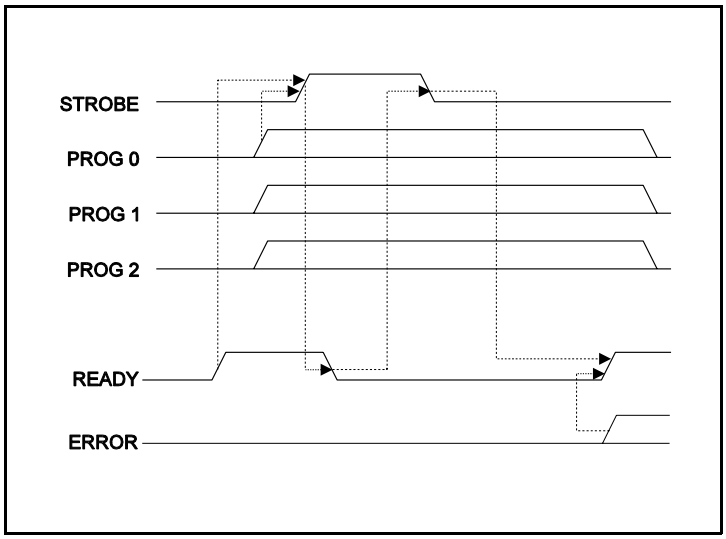


Fig. 19: PLC Mode: Digital Handshake Protocol

The command sequence is in detail:

1. The PLC has to wait until the IPP shows the **READY** signal.

2. In the next step the PLC has to set up the program information **PROG0** to **PROG2** at **INPUT1**, **INPUT 2** and **INPUT 3** of the IPP.

3. The following **STROBE** tells the IPP to evaluate the program information and to execute the corresponding command. Please be sure that the **STROBE** input is set only after one PLC cycle delay. This is necessary because PLC outputs are not guaranteed to switch exactly synchonous.

4. With the leading edge of the **STROBE** signal the IPP reads the program information and starts the motion command. As a handshake signal the IPP will set **READY** to 0 at this time.

5. When the PLC recognizes READY being switched to 0 then it has to set STROBE back to 0 as a handshake signal if the PLC program information corresponds to
   - relative positioning command
   - absolute positioning commands and
   - initialization commands.

   Please refer to topic 7 below how to handle “free run” commands with the PLC.

6. When the IPP has finished the motion command - including the recovery time - and STROBE is zero again, the IPP will set READY back to a high level to show that the motion is finished and the IPP is ready for the next command from the PLC. Just before the READY signal goes high again, any error condition is shown in the ERROR output. So the PLC should monitor the ERROR output when it detects a low to high transition in the READY signal.

   The ERROR output is set by the IPP if
   - the power stage detected a short circuit condition or
   - the heat sink get to hot (> 85°C) or
   - there has been a power fault with the mains supply or
   - the PLC started a program which is ***not defined*** within the IPP.

7. **Notes:**
   - **READY** will become active only if the **PLC** has reset the **STROBE** signal.
   - The **STROBE** signal may be reset after the **READY** signal has disappeared. The motion command will be executed independent of the **STROBE** signal except for the special command **“free run”**. **Free run** will be executed as long as the **STROBE** signal is high, the motion will stop immediately when **STROBE** goes back to 0.

Signal Name	**Meaning**	**Description**	**Comment**
Digital Inputs and Output			
24 V I/O	Auxiliary power supply for the digital output	24 V to be used for the both digital outpus.	Outputs 0 and 1 and the inputs 0,1,2 and 3 are optically isolated to the other circuits within the IPP.
Output 0	Digital output, function in PLC-mode: **READY**	Will connect **24 V I/O** with the output pin if active	
Output 1	Digital output, function in PLC-mode: **ERROR**		
Input 0	Digital input, function in PLC-mode: **STROBE**	PLC compatible input, common return is **0 V I/O**, threshold voltage is about 5 V	With the leading edge of the **STROBE** signal the program information is accepted by the IPP, if **READY** is active at this time.
Input 1	Digital input, function in PLC-mode: **PROG 0**		program code bit 0 (has the value 1)
Input 2	Digital input, function in PLC-mode: **PROG 1**		program code bit 0 (has the value 2)
Input 3	Digital input, function in PLC-mode: **PROG 2**		program code bit 0 (has the value 4)

0V I/O	Common return for the digital inputs		

7.2 Programming of PLC Functions

Before the PLC can work together with the IPP, the motion commands to be associated with every PLC program information have to be defined. To do this please use the software IPCOMM.

In the main axis window please activate the button **PLC** to open the PLC dialog box:

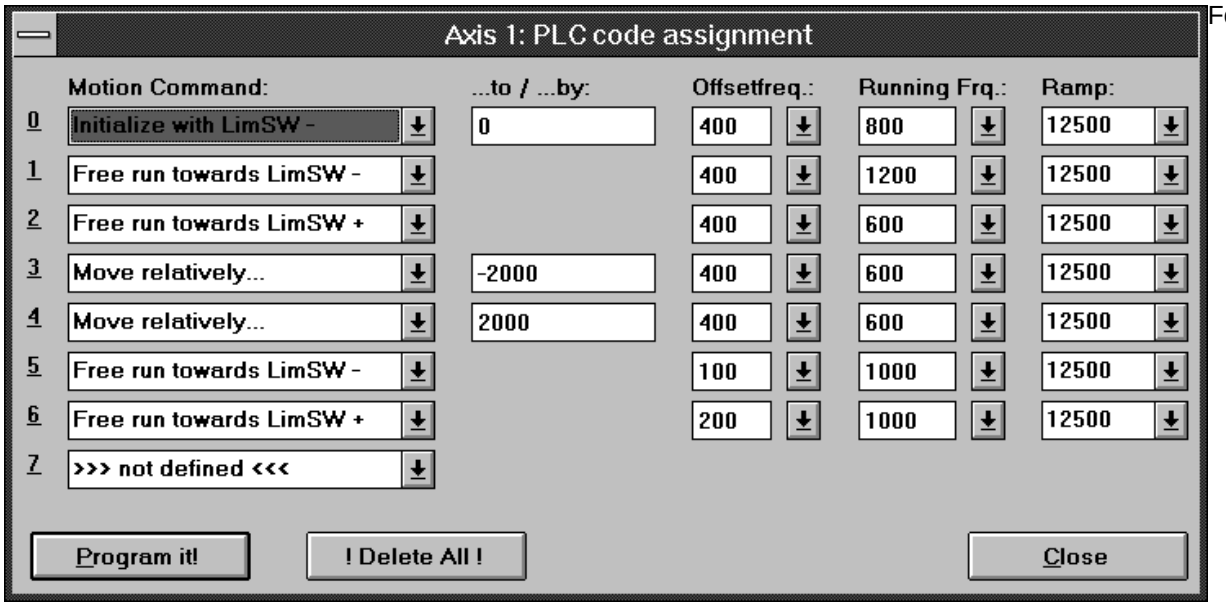

Axis 1: PLC code assignment
Motion Command:
...to / ...by:
Offsetfreq.:
Running Frq.:
Ramp:
0 Initialize with LimSW - 0 400 800 12500
1 Free run towards LimSW - 400 1200 12500
2 Free run towards LimSW + 400 600 12500
3 Move relatively... -2000 400 600 12500
4 Move relatively... 2000 400 600 12500
5 Free run towards LimSW - 100 1000 12500
6 Free run towards LimSW + 200 1000 12500
7 >>> not defined <<<
Program it!
! Delete All !
Close

Fig. 20: Dialog box for programming the PLC functionality of the IPP

For each individual PLC program you will see:

List box **Motion command**
Here you can select the basic command type (reference run, go relative, go to absolute position and so). Depending on your selection in this field IPCOM will show additional fields in the same row where the command specific parameter values can be set. For example, the command **free run** does not need any destination position, so this field will disappear if you select **free run**.

Field **..to / ..by**:
Destination position (if go to absolute position is used) or relative distance to go (if go relative is used).

Field **Offsetfrequ.**:
Here the start/stop frequency can be entered for the specific motion command.

Field **Running frq.**:
Here the running frequency can be entered for the specific motion command.

Field **Ramp**:
Here the acceleration ramp to be used with the specific motion command can be selected.

Note: If you have defined any PLC funtionality and later decide to change the ramp table in the IPP, you will have to re-enter and re-program all of the PLC commands again. Because the PLC commands are based on a valid ramp table, there is no possibility to change the ramps without deleting the PLC information at the same time.

Button **Delete All!**:
Here you can reset all PLC codes.

Button **Program it**:
With this button the PLC codes are written permanently into the IPPs memory, and then they will stay until you change it explicitly again.

Warning:
- **If you leave the dialog box, you will be asked to program the PLC command codes. If you deny this, all input entered for PLC commands is lost!**
- **The PLC commands come into action only if the IPP is in the PLC-mode. Please do not forget to activate this mode, and to make this activation permanent by using *Actual Parameters* in the main axis window!**

Button **Test**:
With this button you may execute the just defined PLC command sequence for testing purposes remotly with your mouse. The button is only active if the command is completely defined. Also in the PLC test mode there are only the buttons active which may be operated at this time.

Field **PLC protocol simulation**:
In this field the actual states of the digital inputs and outputs are shown resp. can be set. Thew signals sent from the PLC to the IPP are displayed, and depending on this signale you may activate only the corresponding PLC command (you will see that only one TEST button is active). To signal the PLC the end of the test, you can set **READY** and **ERROR** output manually with your mouse. The simulation has to be activated with the **START** button and can be aborted with the **STOP** button at any time.

7.3 Hints for Turning the PLC Mode into Operation

Please remind the following hints if working with the IPP's PLC-mode:

1. **The PLC command sequences are available when they are programmed into the IPP's memory. If you abort the PLC dialog box without programming them, any changes are lost and not recoverable.**

2. **If the command sequences are programmed, the IPP is probably not ready to work with them. You have to set the IPP to the PLC mode to have the command active. As a measure of safety otherwise the STROBE input is ignored.**

3. **If you want the IPP set to PLC mode permanently - and not for a fast test - you also have to write the PLC-mode setting into the parameter memory of the IPP. Do this by pressing the button *Actual Parameters* in the main axis window. Settings saved in this way will also be active after a new power up sequence.**

8 Problem Solving

- Under review -

9 ESD-Protection Measures

Every Phytron product delivered to a customer has been well tested.

To prohibit any damages due to electrostatic discharges while handling and manipulating the products, a lot of protective measures have been taken at Phytron.

If you use the product, you have to respect the basic principles of protection measures, too.

Warning:

Please respect measures against electostatic discharges as defined e.g. in CECC 00 015 Ed. 1 if you are handling with Phytron products or you are turning them into operation!

If - for any reason - you want to send the Phytron products back to Phytron, the please use some antistatic bag.

Phytron will take no responsibility for any faults occurring if these recommendations hae been neglegted.

10 Quality Management System

Phytron-Elektronik GmbH has been certified according to DIN/ISO 9001 by the Europäisches Institut zur Zertifizierung von Qualitätsmanagementsystemen EQZERT on November 11th, 1994.

This certification attests an effective quality management system through design and development, production, installation and service.

11 Options

The housing of the IPP series is provided for inserting one extension board and an optional step failure indication (SFI) module.

The SFI module allows the supervision of each stepper motor motion, if the motor is equipped with an appropriate incremental encoder.

The extension board can be customer specific to meet the special needs for more complex I/O requirements. As standard extension board Phytron offers up to now an analog/digital converter board and an digital I/O extension board. Please remind that there is only place for one extension board in each IPP.

11.1 Step Failure Indicator Module SFI

- Under review -

11.2 Analog/Digital Converter Extension

- Under review -

11.3 Digital I/O Extension

- Under review -